Maintenance Manual for Land Information System

Submitted under Task Agreement GSFC-CT-2

Cooperative Agreement Notice (CAN) CAN-00OES-01

Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences

Version 1.1

Revision history:

Version	***Summary of Changes***	***Date***
1.0	Initial Release – Milestone K submission	11 Feb 2005
1.1	Milestone K revisions	02 Mar 2005

Table of Contents

List of Figures

Acronyms and Terms

AGRMET: Agriculture Meteorology Model

AVHRR: Advanced Very High Resolution Radiometer

CGI: Common Gateway Interface

CLM: Community Land Model

CMAP: Climate Prediction Center Merged Analysis of Precipitation

COLA: Center for Ocean-Land-Atmosphere Studies

CVS: Concurrent Versions System

DODS: Distributed Ocean Data System

ECMWF: European Center for Medium-Range Weather Forecasts

FAO: Food and Agriculture Organization

HTTP: Hypertext Transfer Protocol

GDAS: Global Data Assimilation System

GEOS: Goddard Earth Observing System

GrADS: Grid Analysis and Display System

GRIB: Gridded Binary

LDAS: Land Data Assimilation System

LE: Land Explorer

LIS: Land Information System

MODIS: Moderate Resolution Imaging Spectroradiometer

NOAH: National Centers for Environmental Prediction, Oregon State University, United States Air Force, and Office of Hydrology Land Surface Model

NCEP: National Centers for Environmental Prediction

VIC: Variable Infiltration Capacity

1 Introduction

This maintenance manual is a reference document for the Land Information System (LIS). LIS is a project to build a high-resolution, high-performance land surface modeling and data assimilation system to support a wide range of land surface research activities and applications.

This document has been prepared as part of Task Agreement GSFC-CT-2 under Cooperative Agreement Notice CAN-00-OES-01 Increasing Interoperability and Performance of Grand Challenge Applications in the Earth, Space, Life, and Microgravity Sciences, funded by NASA's Earth Science Technologies Office Computational Technologies (formerly High Performance Computing and Communications) Project.

1.1 Purpose and goals

This document serves as the maintenance manual for the Land Information System (LIS). It describes the procedures for installation and maintenance of the subsystems of LIS.

1.2 Scope

This document covers the maintenance of the Land Information System components, which can be categorized specifically as:

- LIS driver (and land surface models)
- LIS user interface
- LIS data management

1.3 Applicable Documents

Several documents are referenced in this document:

- Software Design Document
  http://lis.gsfc.nasa.gov/Documentation/MilestoneJ/swddJ3-3.pdf
- Data Management Design Document
  http://lis.gsfc.nasa.gov/Documentation/MilestoneJ/dmJ3-0.pdf
- User Interface Design Document
  http://lis.gsfc.nasa.gov/Documentation/MilestoneJ/UI.pdf
- Interoperability Design Document
  http://lis.gsfc.nasa.gov/Documentation/MilestoneJ/iop_design.pdf
- User's Guide
  http://lis.gsfc.nasa.gov/Source/LIS3.0/usersguide.pdf
- Developer's Guide
  http://lis.gsfc.nasa.gov/Source/LIS3.0/devguide.pdf
- Documentation of LIS system components
  http://lis.gsfc.nasa.gov/Source/LIS3.0/code_doc.html
- Concurrent Versions System Guide

/data1/cvsroot on IO node X2 of the cluster

2 Environment

The LIS project utilizes three systems to do high-performance, high resolution land surface modeling, and store and distribute data: 1) the web server, also known as hsb; 2) the public data server, also known as lisdata; and 3) the cluster, which is used as the test bed for high-resolution global modeling. The system administrators of Code 974 maintain these systems. Contact sysadmin@hsb.gsfc.nasa.gov for any questions or concerns regarding these systems.

The specifications of these systems are as follows:

Name	Operating System	Processor	Memory
HSB	Red Hat Enterprise Linux WS release 3	Intel Pentium 4 2.8 GHZ	1 GB
Lisdata	Debian Linux 3.0	AMD Athlon 2.0 GHZ	4 GB
Cluster IO nodes	Red Hat Linux 7.3	Dual AMD XP 2000	2 GB Double Data Rate
Cluster compute node	Red Hat Linux 7.2	AMD XP 1800	1 GB

2.1 HSB

The hsb machine is used as the web server for LIS. Hsb uses Apache 2.0.46 as its web server. It houses all of the web pages for the LIS web site, and uses these web pages to serve as the front-end for the LIS user interface.

2.2 Lisdata

The lisdata machine is used as the public data server for LIS. The public GrADS-DODS Server and the Land Explorer are run from the lisdata machine, with links from the LIS web site.

2.3 The Cluster

The cluster serves as a test bed for LIS runs. The cluster has 200 computers in total. Eight of them are the IO nodes, or "queen" nodes. The remaining 192 computers are compute nodes. Overall, the LIS cluster has 208 AMD XP processors of 1.53 GHz and above, 112 GB of memory, 21 TB of disk space, 192 fast Ethernet connections, and 10 gigabit Ethernet connections. Please refer to “Optional Milestone F” of the LIS

Milestones for further documentation of the LIS cluster.
http://lis.gsfc.nasa.gov/Documentation/Documents/cluster.shtml.

3 LIS software

The LIS software system consists of a number of components: (1) The LIS driver, the core software that integrates the use of land surface models, data management techniques, and high performance computing; (2) Community land surface models such as CLM, Noah, and VIC; (3) A web-based user interface that is used to distribute the LIS source code and data, along with visualization tools such as the Land Explorer; and (4) Data management and distribution tools such as the GrADS-DODS server

3.1 LIS source code

The LIS source code consists of the LIS driver; the CLM, NOAH, and VIC land surface models; and the LIS scripts, which are used to compile and build the driver. They are required to run LIS, along with the Message Passing Interface (MPICH) version 1.2.5.2, which can be obtained at http://www-unix.mcs.anl.gov/mpi/mpich/. Other optional scripts are available from the LIS web site to perform a run using the GDS data management system or to process the output data for visualization.

The code consists of Fortran 90 and C source files along with several libraries. The compilation of this source code is managed via a makefile based on GNU's gmake. The steps to obtain, compile, run, and maintain the source code is completely spelled out in the User's Guide, beginning at Section 3. For coding and documentation conventions or development, or extending the code to add new models, forcing schemes, domains, or input data, please refer to the Developer's Guide.

LIS source code is archived in a Concurrent Versions System repository, using CVS version 1.11.1p1. The CVS administrator performs all updates and backups of the repository. Please refer to the CVS guide for more details.

3.1.1 LIS driver

The core of the LIS software system is the LIS driver, which controls program execution. The LIS driver is a model control and input/output system (consisting of a number of subroutines, modules written in Fortran 90 source code) that drives multiple one-dimensional land surface models. The one-dimensional land surface models apply the governing equations of the physical processes of the soil-vegetation-snowpack medium. These land surface models aim to characterize the transfer of mass, energy, and momentum between a vegetated surface and the atmosphere.

3.1.2 Land Surface Models

CLM (Community Land Model) is a 1-D land surface model, written in Fortran 90, developed by a grass-roots collaboration of scientists who have an interest in making a

general land model available for public use. LIS currently uses CLM version 2.0. The source code for CLM 2.0 is freely available from the National Center for Atmospheric Research (NCAR) (http://www.cgd.ucar.edu/tss/clm/)

The Community Noah Land Surface Model (N: NCEP; O: Oregon State University, Dept. of Atmospheric Sciences; A: Air Force (both AFWA and AFRL - formerly AFGL, PL); and H: Hydrologic Research Lab - NWS (now Office of Hydrologic Development)) is a stand-alone, uncoupled, 1-D column model freely available at the National Centers for Environmental Prediction (NCEP). LIS currently uses NOAH version 2.6. ftp://ftp.ncep.noaa.gov/pub/gcp/ldas/noahlsm/

Variable Infiltration Capacity (VIC) model is a macroscale hydrologic model, written in C, being developed at the University of Washington and Princeton University. The VIC code repository along with the model description and source code documentation is publicly available at the Princeton website. LIS currently uses VIC version 4.0.3. http://hydrology.princeton.edu/research/lis/index.html

3.1.3 LIS Interoperability

LIS consists of an integrated set of components and routines organized into several functional abstractions to facilitate development of land surface modeling applications. The components in LIS are implemented using object oriented design principles. LIS integrates these components into a framework facilitating their effective use in solving a variety of different problems. LIS provides extensible features that enable the reuse of LIS structure as well as the use of standards and tools proposed by other earth system modeling groups. Please refer to the Interoperability Design Document for further details of LIS interoperability.

3.2 LIS User Interface

The LIS user interface consists of a web site for accessing code, data and documentation; visualization tools, such as the Land Explorer; and configuration tools, such as the Card File Generator.

The user interface subsystem takes a typical multi-tier client-server system architecture. On the client side, a user has two types of client programs to use as the front-end: a web browser, or a DODS client program. On the server side, a general-purpose web server is used to serve clients with a web browser, and a GrADS-DODS server is deployed to serve DODS clients. Besides these components, CGI scripts and CGI-GrADS gateway scripts are used as the middleware to perform dynamic processing based on users' interactive requests sent through web browsers. The user interface of LIS is an important component of LIS that allows the interactive, flexible use of the LIS software to users. The LIS user interface is web-based, and allows for cascading complexity depending on the level of user's need to control the system. Figure 1 shows the user interface architecture.

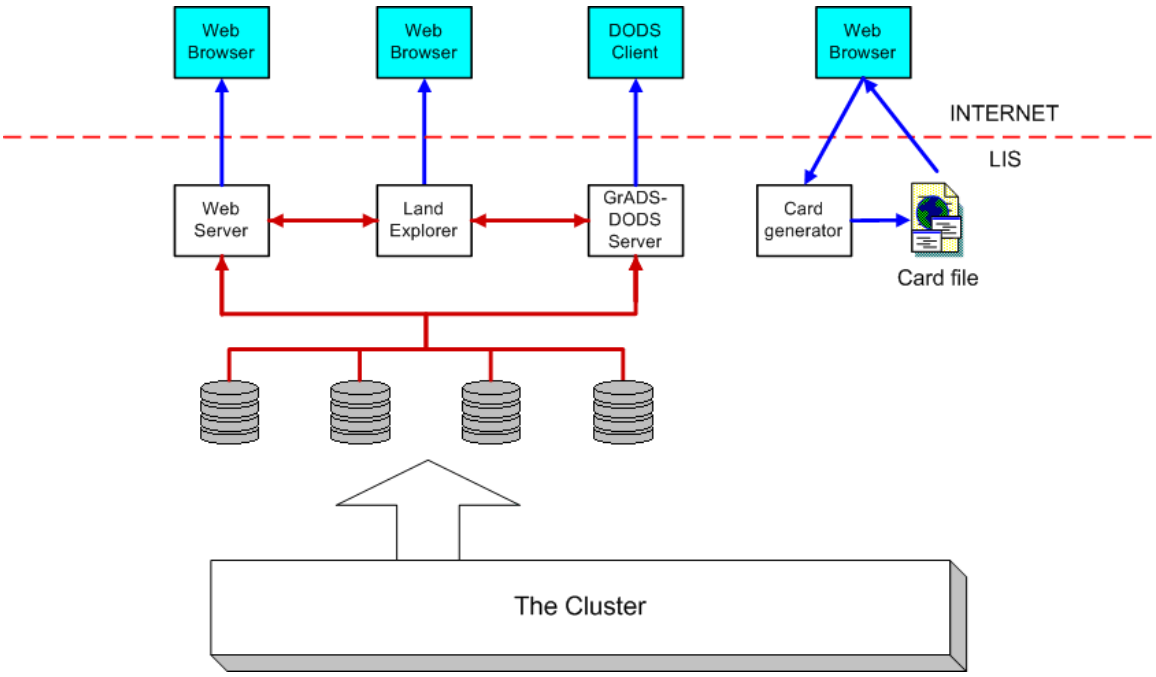

Figure 1: LIS user interface architecture

This diagram illustrates the levels of data access. The first arrow from the left represents users accessing static data using a Web browser. The remaining arrows represent users who access LIS data by either of the following methods: 1) by using a Web browser to dynamically generate images and manipulate data; or 2) by using a DODS client to manipulate a subset of data.

Users that register with LIS will be able to access raw data by HTTP download. Registered users can also use a web browser to obtain a customized copy of the LIS configuration file to set up a LIS model run on their local machine, after the raw data has been downloaded. Both sets of users (registered and non-registered) are able to access the LIS user interface via a web browser, with an entry page as shown in Figure 2.

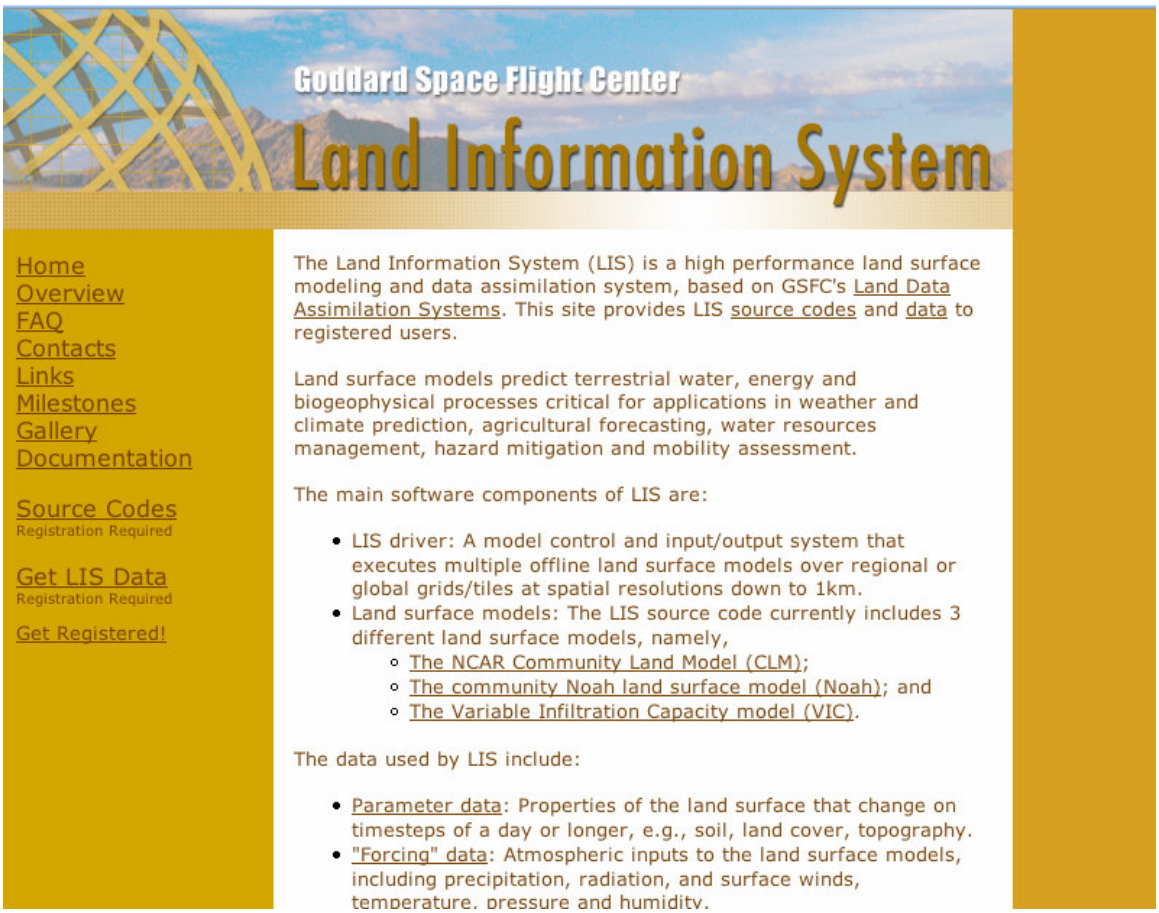

Figure 2: LIS home page

3.2.1 LIS Web Pages

3.2.1.1 Format and Layout

The LIS web site is housed on the hsb machine in the directory /www/lis. It utilizes two style sheets, which are used to control the appearance and layout of the web site. The first style sheet gives the layout as shown in Figure 2. This design is used for the LIS home page and the other "main" pages that are accessed by clicking on the navigation bar on the left panel. It uses two server-side includes (SSI). The first SSI shows the panels on the left and right sides of the page, and the title bar. This file is /www/lis/gnav1.html. The second SSI is used for the footer information at the bottom, which shows the curator, NASA authorizing official, and security messages. The curator's and authorizing official's information are coded in hexadecimal format, which is discussed in greater detail in the "Contacts" section. This footer file is /www/lis/gnav2.html. These are placed inside of the html between the <body> tags, at the same location in each file. Here is an example of the usage of the Server-Side Include on the LIS home page.

```
<HTML>
<HEAD>
<TITLE>NASA :: Goddard Space Flight Center : Land Information Systems</TITLE>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
</HEAD>
<BODY BGCOLOR=#FFFFFF LEFTMARGIN=0 TOPMARGIN=0
MARGINWIDTH=0 MARGINHEIGHT=0
alink="#663300" link="#663300" vlink="#663300">
<font color="#663300"></font>
<!--#include file="gnav1.html"-->
```

HTML for the web page……..

```
<!--#include file="gnav2.html"-->
</BODY>
</HTML>
```

NOTE: Only the homepage will use the syntax <!--#include file=”gnav1.html>. All of the other “main” pages will use the syntax <!--#include virtual="/gnav1.html"--> The gnav files are local files in the home directory, but in the other directories the “include virtual” must be used because the files are not local to that directory.

A second style sheet is used on the web site for the gallery, data, and source code pages, as shown in Figure 3. This style sheet is used to allow for more text and data to be entered on the pages without the restrictions of the panels on the main pages. This format uses a title bar at the top of the page, with some text underneath. The path for this style sheet is /css/lis.css, and it does not use a server-side include. It uses the <link> tag, which appears at the end of the <head> section in the HTML.

```
<link rel="stylesheet" href="/css/lis.css" type="text/css">
```

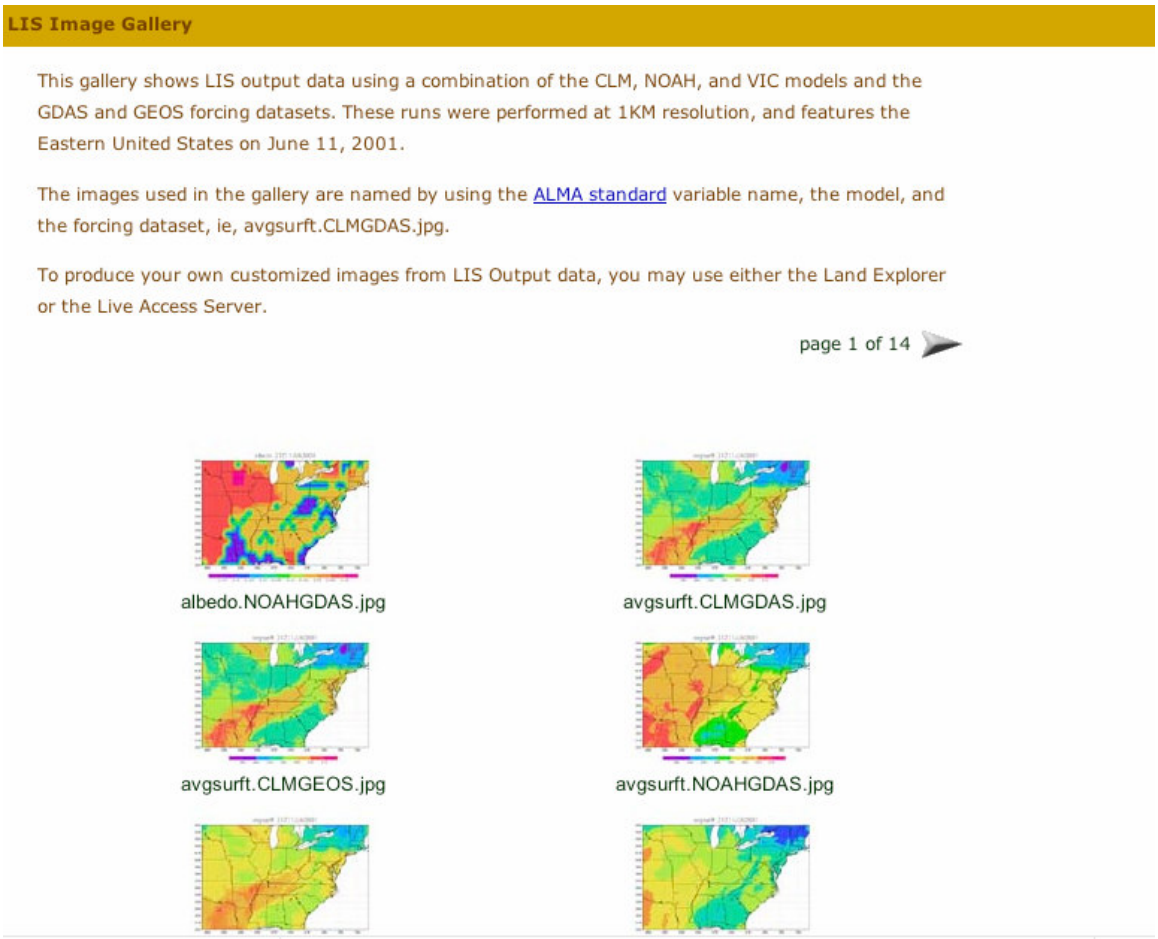

Figure 3: LIS Gallery

3.2.1.2 Directories and Files

The LIS home directory contains nine subdirectories that organize LIS information, data, and source codes. These directories are:

Overview: A short overview of the LIS project and its origins.

Links: Links to research related to LIS

Gallery: Images of LIS output data and links to the LIS visualization engines. The gallery was created using Adobe Photoshop. To add images or to create a new gallery, follow the steps outlined in this document. http://www.adobe.com/digitalimag/tips/phsel2webgallery/pdfs/phsel2webgallery.pdf

Contacts: This directory contains the name and contact information for contributors to LIS. The e-mail addresses of the contributors are coded in hexadecimal format instead of using the conventional <mailto> tag, in order to

protect against spam. Some resources on the topic: (from http://www.google.com/search?q=mailto+link+spam&ie=UTF-8&oe=UTF-8)

http://www.bris.ac.uk/is/services/computers/nwservices/mail/hidden-mailto.html
http://www.hiveware.com/enkoder.cgi
http://webdesign.about.com/library/weekly/aa080301a.htm

FAQ: This directory's default page answers questions concerning LIS data and resources, registration, platform support, and output format. There are also input data documentation pages, params.shtml, and forcing.shtml, included in this directory.

Milestones: Each file in this directory refers to a corresponding milestone (i.e. A.shtml for Milestone A). Each file lists the deliverables associated with that particular milestone.

Documentation: Each subdirectory's name refers to a corresponding milestone. These subdirectories contain the required documents for each milestone. There is also an archive subdirectory for draft versions of documents.

Source Code: The source code subdirectories are named by the version of LIS, i.e. LIS 3.0, LIS 3.1. These subdirectories contain 1) Source files, which include the LIS driver, the land surface models, libraries, science routines, and modules; 2) Scripts, which include the card file, the compile and build script, the post-processing scripts, and OpenDap scripts. Section 5 of the User's Guide lists all of the files and scripts that are needed for downloading.

Data: Contains links to LIS parameter and forcing data. Also contains scripts that were used to generate the LIS data from the original datasets.

Non-registered users are the general public, who access the LIS data through a standard web browser. Information provided to this class consists only of static images and text. This static content is served via the web server. This group of users does not have direct access to the data and their usage of system resources is very limited. Therefore, for this class of users there are no additional authentication or authorization procedures. Data provided to non-registered users is as follows:

3.2.1.3 Registration

Registered users will have more direct access to LIS data. They will accomplish this by fetching directly through an HTTP download. The GrADS-DODS server provides the users with the ability and flexibility to get only a sub-set of the data they need. If the entire data set is desired, users are able to directly download the raw data.

Internet users can register by filling out the form on the LIS web site. After a user fills out the form on the registration page, the script "lis_reg2.cgi" is launched, which parses the form and stores the username, email address, password, and the date of registration.

This script also writes the username and password in the .htpasswd1 file, so that it will allow the new user to login in and access the protected areas. Lastly, a confirmation email is sent to the new user. Registration is used to allow an extra layer of security between the Internet user and the LIS hardware/software, and also to keep an internal record of LIS users.

Apache's .htaccess feature is used to protect the Source Code and Data directories of the web site. Users must register in order to access these areas. Here is an example of the .htaccess file.

AuthUserFile /www/lis/.htpasswd1
AuthName "Restricted "
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>

"Require valid-user" allows all registered users to access this directory. To protect a new directory, the web site administrator should place a copy of the .htaccess file in the directory that is to be protected. The administrator can then allow all registered users to access this directory by using the "require valid-user", or he can only allow specified users to the new directory. If this specified user is already a registered member, only the user name is required.

For example, to protect a directory that will allow user lis and user ldas to access the contents, the .htaccess file should be edited like this:

AuthUserFile /www/lis/.htpasswd1
AuthName "Restricted "
AuthType Basic
<Limit GET POST>
require user lis
require user ldas
</Limit>

The administrator can allow as many users as he wishes to access the directory by listing the users between the <Limit> tags. If the administrator wishes to protect a directory with a new username/password combination, the htpasswd command is used:

htpasswd /www/lis/.htpasswd1 newuser

The system will then ask for a password for "newuser". After this is given, edit the .htaccess file in the directory to be protected:

AuthUserFile /www/lis/.htpasswd1
AuthName "Restricted "

AuthType Basic
<Limit GET POST>
require user newuser
</Limit>

3.2.1.4 The LIS Card File Generator

The Land Information System Card File Generator allows registered users to customize a LIS model run. The LIS "card file" is a file that the LIS driver uses to set the parameters for a run. Some of these parameters are modifiable, and the Card File Generator allows users to modify these parameters using an HTML form, in conjunction with a CGI perl script and a template file.

The CGI script, “writecard.cgi”, parses the user’s parameters. The template file is a copy of the LIS card file with keywords surrounded with double percent symbols (%%KEYWORD%%). These keywords represent values that will be replaced by the user’s selected values. The CGI script then creates the output file, lis.crd, which is a copy of the template file with the user’s selected values in place of the keywords from the template. The CGI script then points the browser to the lis.crd file. This lis.crd file is a temporary copy located on the http server. The user must use the browser's "save page" feature (or copy/paste) to obtain a copy of the file.

An on-line tutorial is also available to help users. The LIS configuration web interface is shown in Figure 5.

The card file parameters are as follows:

Land Surface Model, consisting of CLM, VIC, or NOAH.

Spatial Resolution, consisting of 1km, 5km, 1/4 degree, 1/2 degree, 1 degree, and 2 by 2 and 1/2 degree.

Base Forcing data, consisting of GEOS or GDAS data.

Observed Forcing data, consisting of precipitation (CMAP) and radiation (AGRMET).

Temporal Domain, where the year, month, day, and time for the run are specified.

Grid Type, which has the types of grids supported by LIS.

Spatial Domain. Users can select the Parameter and the Execution Domain for their run.

LAI from MODIS, AVHRR, or user selected (Table-based) data.

Landcover, consisting of MODIS or University of Maryland AVHRR data.

Soils, consisting of FAO, Statsco, or user defined data.

Elevation, consisting of GTOPO30.

Run-time options include restart capability, elevation correction, the ability to set the number of tiles per grid and the min % of a tile.

Output options include data format (binary or grib), writing output to a tile domain or grid, writing output to a single file or separate files, and the writing or forcing and parameter data (LRD, Sec 8.41 –8.42).

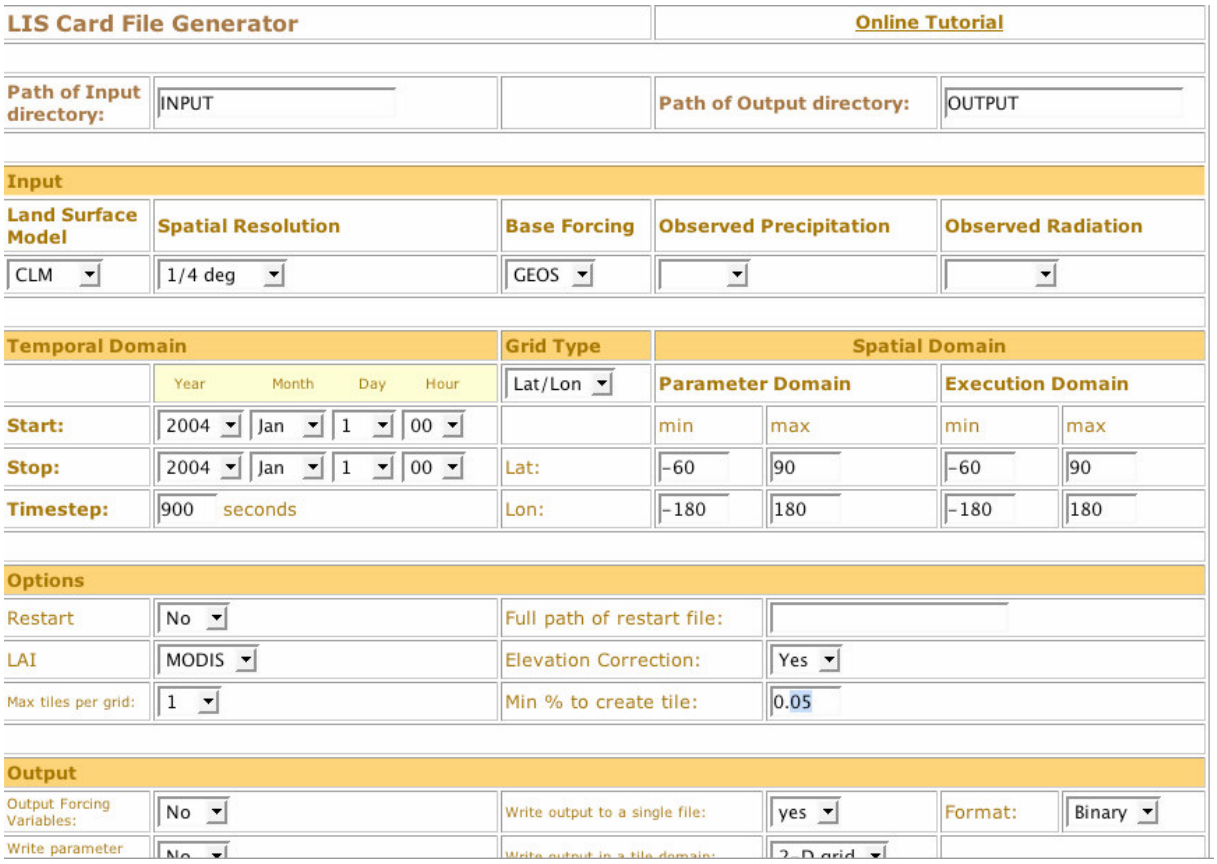

Figure 4: LIS Card File Generator

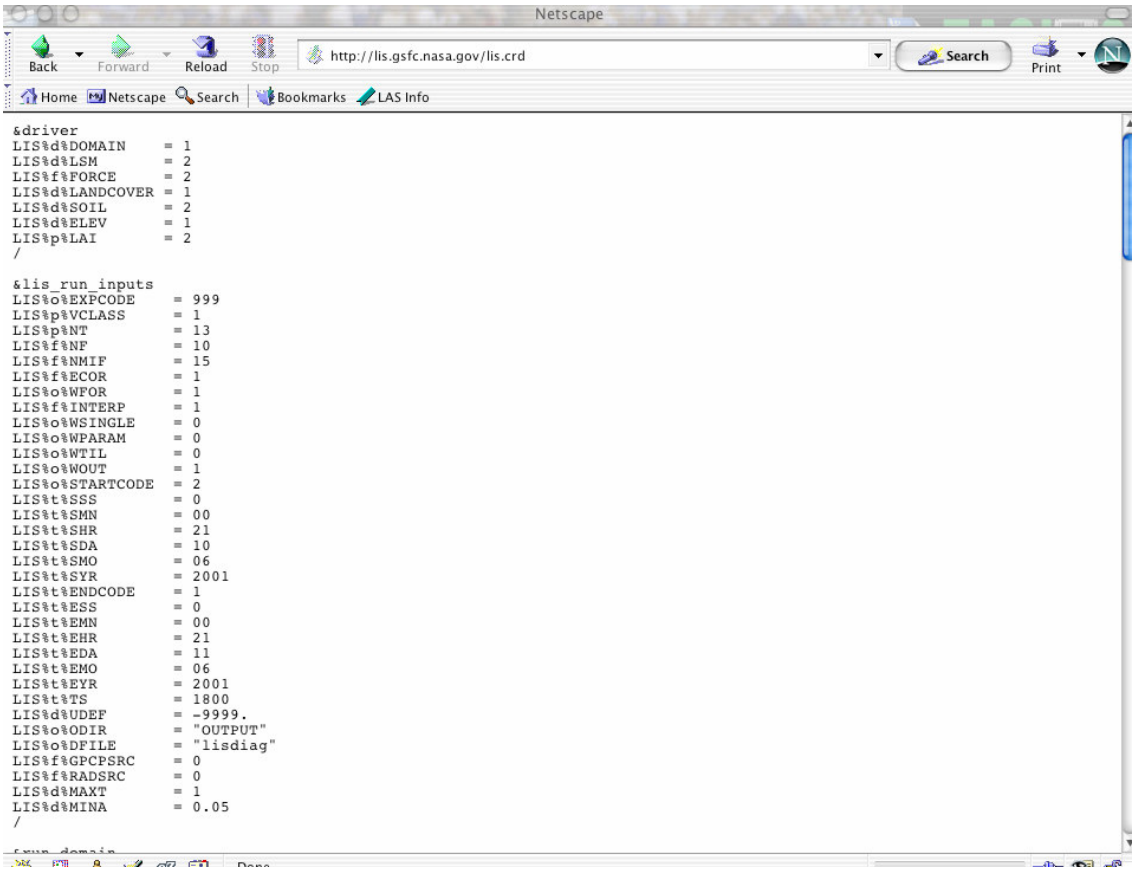

Figure 5: LIS Card File

3.3 Visualization

To visualize LIS data, a web-based visualization system called "Land Explorer" (LE) was developed.

3.3.1 Land Explorer (LE)

LE is tightly integrated with the LIS GrADS-DODS server. It is designed to let users interactively visualize and explore LIS data at all resolutions, featuring an intuitive web interface and fast response. No special requirements, such as javascript, cookies, java applets, or pop-up windows are needed. Users can access LE by using a web browser to access the LIS GrADS-DODS server, located at http://lisdata.gsfc.nasa.gov:9090/dods/. Each variable has a “Visualize” link that allows users to use LE for that specific variable.

LE adds a web-based interactive visualization capability to GrADS-DODS server (GDS) 1.2.8. It contains a patch for GDS' source code, a perl CGI script and a few other files.

Steps to install the LE:

Requirements:
* GrADS-DODS server v1.2.8
* A web server with cgi support (e.g., Apache)
* Perl
* GrADS 1.8 or newer with DODS support.

* Note - LE can not be publicly distributed because of licensing issues with COLA. The tar file is available from the lisdata machine, /home/gds/gds-extension.tar, to only be used to visualize LIS GDS data.

1. Locate the web server document root directory, $WEBROOT, and copy all the files in the "images" sub-directory to $WEBROOT/images/.

2. Locate the web server's cgi directory, and copy the following three files there:
gds_title.gs, gdsplot.pl, gdsplot.template

3. Edit gdsplot.pl in the cgi directory to update the values of the variables $GADDIR, $GASCRP and $GRADS to reflect your environment, and make sure the file has executable permission by the web server.

4. Edit gdsplot.template in the cgi directory to update all the links to the cgi "gdsplot.pl".

5. Locate GDS 1.2.8's root directory, $GDSROOT, and copy "GradsGridExtracter.java" to $GDSROOT/src/org/iges/grads/server/.

6. Edit the copied "GradsGridExtracter.java" to update lines 603 and 604 shown below to fit your system setting. Change "http://lisdata.gsfc.nasa.gov/las-bin" in line 603 to your webserver's cgi path, and http://http://lisdata.gsfc.nasa.gov:9090/dods/ in line 604 to your GDS server's path.

```
603     info.print("<a href=\"http://lisdata.gsfc.nasa.gov/las-bin/gdsplot.pl?var=" +
604     lvar + "&url=http://lisdata.gsfc.nasa.gov:9090/dods/" +
```

7. Recompile the source code and restart the GDS server:
```
cd $GDSROOT/src/
./makejar
cd ..
./rebootserver
```

Then the GDS will be running with the visualization capability.

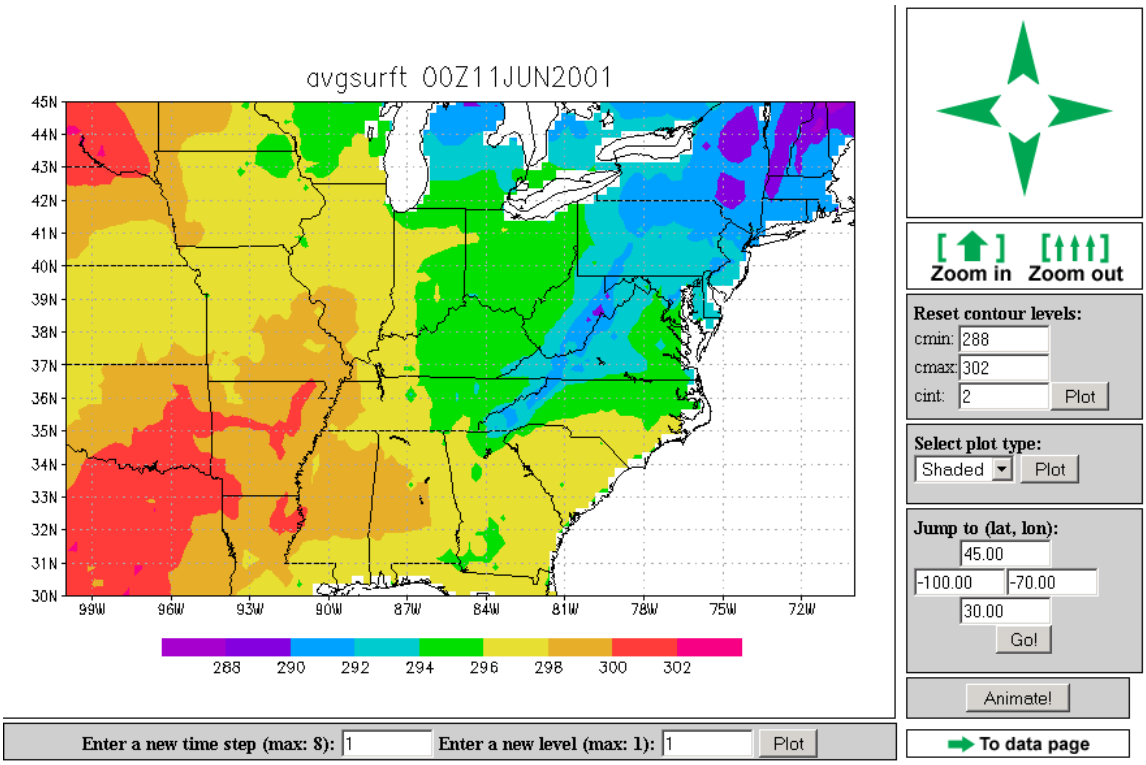

Figure 6: Land Explorer (LE)

4 Data Management, Distribution, and Storage

LIS is designed to perform land surface simulation and data assimilation on parallelized computing platforms, at very high spatial resolutions (up to 1km) and in near real-time. It imports a continuous flow of atmospheric forcing data and a collection of land surface parameter datasets, and produces a huge amount of land surface data to satisfy the needs of diverse users. Such an operation poses many challenges to the data handling functionality of LIS, and requires a highly reliable and efficient data management design. Specifically, the design covers the following five functional areas:

- End-to-end data flow
- Data retrieval, distribution and storage
- Data analysis capabilities, including interpolation, re-projection, sub-setting, and file format conversion
- Link to the user interface
- Interoperability through ALMA and ESMF compliance

Please refer to the Data Management Design Document for a complete layout of the LIS

Data Management Design.

4.1 GrADS-DODS Server

The LIS project utilizes the GrADS-DODS server technology to manage and distribute data. LIS has a public GDS to make data available to users who need direct access to the data in near real-time using desktop software. Users who are utilizing a DODS client (GrADS, Matlab, IDL, etc.) are able to directly manipulate subsets of LIS data through the DODS protocol. The public GrADS-DODS server is located on the lisdata machine, in the directory /home/gds. Once logged in, the GDS administrator must log in as user gds to perform administrative duties. Contact sysadmin@hsb.gsfc.nasa.gov to obtain the password for user gds.

This section discusses how to install the GDS and add new datasets to the configuration file. Further GDS documentation can be found in the Administrator's Guide, at http://grads.iges.org/grads/gds/doc/admin.html.

4.1.1 Installation

Requirements

- The GDS can run on any UNIX platform for which both Java and GrADS are available.

- Java Virtual Machine (JVM) that supports Java 1.3 or higher. Enter java -version at the Unix command prompt to find out what JVM is installed on your system. The Java Virtual Machine is a free download either from Sun Microsystems (http://java.sun.com), or your operating system manufacturer's website.

- GrADS. Because the server uses some new features in GrADS, version 1.8 or higher is needed. Handling station data and client uploads requires version 1.9 or higher. The latest version of GrADS is available at the GrADS home page (http://grads.iges.org/grads/grads.html).

Download and setup

The latest version of the GDS is available at the GDS home page (http://grads.iges.org/grads/gds/index.html) as a compressed tar archive.

Root user access is not needed to run the GDS. There is no build or system install process, because it is a cross-platform Java application. And any number of GDSes can be run on the same system, as long as they are configured to use different ports.

After unpacking the archive, edit the configuration file and tell the GDS where to find GrADS, by editing the <invoker> tag. If you are not using a full GrADS distribution,

make sure the GrADS executable you specify is capable of opening the types of dataset you wish to use.

Next, double check with other users and/or your system administrator, to make sure that port settings for the GDS do not conflict with ports that are already in use. By default the GDS uses ports 9090 and 9095.

At this point, you should be able to start the server and view the example dataset.

If you plan to serve netCDF, HDF, or DODS data, also make sure that the GADDIR environment variable in the GDS startup shell points to the location of the GrADS supplementary data files (available from the GrADS downloads page). In particular, the file udunits.dat must be present in this directory, since it is needed for COARDS metadata processing. If GADDIR is not set, this error will occur:

error: can't import *dataset_name*; metadata extraction failed for *dataset_file*; couldn't open *dataset_file*

when the GDS tries to access any netCDF, HDF or DODS data.

4.1.2 Starting and Stopping the server

There are four scripts in the server home directory that are used to control the GDS, which send brief messages to the terminal, and record their actions in more detail in the file log/console.out:

startserver - Starts a background task that runs and monitors a GDS. It will respawn the GDS if the process dies.

stopserver - Shuts down the GDS, preventing the process from respawning.

rebootserver - Restarts the GDS.

cleanup - Restarts the GDS, clearing all temporary data such as cached metadata, analysis results, and blocked IP addresses.

These invoke the corresponding scripts in the bin directory:

bin/gds-start.sh
bin/gds-stop.sh
bin/gds-cleanup.sh

4.1.3 The GDS configuration file

The GDS configuration file is called gds.xml. This file is configured to set a limit on system resources and perform administrative functions. It is most commonly used to add new datasets to the GDS. If a new file is added, the GDS must then be rebooted for any changes to the gds.xml file to take effect. Further documentation for the GDS configuration file can be found at http://grads.iges.org/grads/gds/doc/tag-ref.html.

To add new datasets to the GDS, first make sure that all of the datasets are ready to be opened by GrADS. If you have COARDS-compliant NetCDF data, they are ready to go as is. Otherwise, you may need to generate some control and/or map files..

Once this is done, all you need to do to put your datasets online is tell the GDS where they are, using the configuration file. Adding a <datadir> tag inside the <data> tag does this.

```
<gds>

 <catalog>
  <data>

    <datadir name = "Parameter-Data"
             doc  = "http://lis.gsfc.nasa.gov/FAQ/params.shtml"
             file = "/www/LISdata/GDSdata/Parameters"
             recurse = "true"
             suffix = ".ctl"
             format = "ctl"/>

          …….
          ……..
          ……..

  </data>
 </catalog>
 </gds>
```

Parameter Data is the name given to this dataset. Doc is the URL of on-line documentation that is available for this dataset. The raw data is located locally in the directory /www/LISdata/GDSdata/Parameters. The GDS will recurse through this entire directory to find all control files (.ctl) and load the corresponding data under the Parameter-Data directory.

Note: The GDS does not attempt to access datasets until the first time they are requested by a client, so it may not immediately complain about unusable datasets. Test all datasets in a DODS-enabled client.

All of the datasets that are served by the LIS GDS are stored locally on the lisdata machine. The gds.xml file gives the path of these datasets in the <datadir> tag. Here is a list of the datasets currently being served by the GDS:

```
<datadir name = "Parameter-Data"
        doc  = "http://lis.gsfc.nasa.gov/FAQ/params.shtml"
        file = "/www/LISdata/GDSdata/Parameters"
        recurse = "true"
        suffix = ".ctl"
        format = "ctl"/>

        <datadir name = "Elevation"
         file = "/www/LISdata/GDSdata/Elevation"
         recurse = "true"
         suffix = ".ctl"
         format = "ctl"/>

<datadir name = "Forcing"
       doc  = "http://lis.gsfc.nasa.gov/FAQ/forcing.shtml"
       file = "/www/LISdata/GDSdata/Forcing"
       recurse = "true"
       suffix = ".ctl"
       format = "ctl"/>

<datadir name = "Output/MilestoneG/US"
       file = "/www/LISdata/OUTPUT-G/US"
       recurse = "true"
       suffix = ".ctl"
       format = "ctl"/>

<datadir name = "Output/MilestoneG/global"
       file = "/LISdata/Output/1km"
       recurse = "true"
       suffix = ".ctl"
       format = "ctl"/>
       recurse = "true"
       suffix = ".ctl"
       format = "ctl"/>

<datadir name = "Output/MilestoneJ/0.25"
       file = "/DATA/MilestoneJ/0.25"
       recurse = "false"
       suffix = ".ctl"
       format = "ctl"/>
```

4.2 Parameter Data Distribution

LIS derives its high-resolution parameter datasets from native datasets obtained from various sources (University of Maryland Global Land Cover, MODIS LAI, Boston University AVHRR LAI, the Food and Agriculture Organization Soils Data, etc.) Fortran scripts are used to regenerate the data at higher resolutions, and a neighbor search is conducted to fill the missing values. To obtain the original datasets, the scripts used to create the LIS datasets, and the LIS high-resolution data, these steps must be taken:

Go to the LIS web site.
Follow the link for "Get LIS Data".
Follow the link for "Land Surface Parameters" for Milestone G.

The parameter data tables allow users to view the metadata and obtain the DODS url for a dataset by following the "GDS" link. Users can download the raw binary data by following the "HTTP" link. Lastly, users can obtain the processing scripts and view the processing information for the dataset by following the "Info" link. If a link for processing information says "GLDAS", the Land Data Assimilation (LDAS) group must be contacted to obtain the original data sources and processing information. For the LAI datasets, follow the "HTML" link. The LAI data table has links to the raw data and the processing scripts used to generate the data at different resolutions.

4.3 Forcing Data Distribution

LIS runs are driven by atmospheric forcing data. LIS uses two atmospheric data assimilation model outputs as the base forcing: one is the NCEP GDAS output, and the other is the NASA GEOS output. The model forcing data are then superposed by optional observed datasets such as radiation and precipitation, provided by the Air Force AGRMET (Agricultural Meteorology modeling system), and the NCEP CMAP (Climate Prediction Center Merged Analysis Precipitation).

LIS also records the elevation correction, which is the difference between the elevation data LIS uses and an atmospheric model uses. LIS adjusts the forcing data whenever the elevation differs between LIS and the model using Fortran scripts. LIS uses elevation data derived from GTOPO30, while each atmospheric model (GEOS, GDAS, etc) uses its own elevation data. To obtain the forcing datasets, elevation correction data, and the scripts:

Go to the LIS web site.
Follow the link for "Get LIS Data".
Follow the link for "Atmospheric Forcing Data" for Milestone G.

The forcing data table allows the data to be accessed either by GDS or http. The elevation correction can be access by the HTML link.

4.4 Output Data Distribution

A sample of LIS output data is made available to users through the GDS server, which allows interactive visualization and use of the data. It is not available via http download. Images of LIS output data can be viewed on the web in the Gallery.

4.5 Data Storage and Backups

Hsb and lisdata use several ATA IDE disks for their respective file systems. Hsb uses two of these disks, LISD1 and LISD2 for web data storage and distribution. The 1/4 degree resolution data is uncompressed, and is in binary format; the 5KM and 1KM datasets are compressed because of their huge file sizes. They are also in binary.

The input data is categorized as: 1) GVEG (Global Vegetation), which account for the UMD mask and Vegetation classification maps; 2) BCS (Boundary Conditions), which account for the Soils and NOAH specific datasets; or 3) LAI (Leaf Area Index), which uses MODIS and Boston University AVHRR data. Once the data is placed in one of these categories, it is then organized by its resolution (1/4 degree, 5KM, or 1KM). The forcing data is located in the Forcing directory on LISD1. They are distributed as zipped tar files. LISD1 and LISD2 together supply 500GB of disk space. These disks are also cross-mounted onto lisdata.

Lisdata uses an external RAID tower, partitioned as /DATA and /DATA1. This RAID supplies 2.4 TB for LIS data storage and backups. The data that is served by the GDS is stored locally on lisdata. The current locations of the datasets that are served by the GDS can be found in the GDS configuration file, as shown in section 3.3.3. Some datasets that are used by the GDS currently use an NFS mount from the air machine to lisdata, and are accessed through the directory /www/LISdata/GDSdata. This data will eventually be moved onto the /DATA disks for redundancy. These datasets are all uncompressed, with control files for each dataset to allow the GDS to access the data. The LIS web site is backed up on lisdata, at /www/lis.

The cluster has to store a copy of all of the LIS input data on each of the master nodes, in order to distribute pieces of the data to the compute nodes during a LIS run. The cluster has 1.2 terabyte raids that are used to backup each IO node. This provides excellent redundancy for storage and backup purposes. The cluster also has a storage node, which is only used for that purpose. It has 2 RAID disks, which are 1.2 terabytes each. It is partitioned into 4 disks, with 2 disks serving as scratch space for large runs, and the other 2 disks strictly being available for storage. The storage disks are non-visible to users, and are backed up nightly. A clean backup is performed every two weeks for the storage node as well.

5 Points of Contact

Dr. Paul Houser – Code 974 Branch Head and Co-PI
Paul.R.Houser@nasa.gov

Dr. Christa Peters-Lidard – Project Manager and Co-PI
Christa.Peters-Lidard@nasa.gov

Dr. Sujay Kumar – Code developer
Sujay@hsb.gsfc.nasa.gov

Dr. Yudong Tian – Code developer, Data Management, Web Curator
Yudong@hsb.gsfc.nasa.gov

James Geiger – Code developer, CVS Admistrator
James.V.Geiger@nasa.gov

Megan Larko – Code 974 System Administrator
Kevin Miller – Code 974 System Administrator
Aedan Jenkins – Code 974 System Administrator
sysadmin@hsb.gsfc.nasa.gov